

Heroku Vs. AWS(Elastic BeanStalk)

Introduction to Heroku:-

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. The platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

Heroku is fully managed, giving developers the freedom to focus on their core product without the distraction of maintaining servers, hardware, or infrastructure.

The Heroku Platform

Heroku is a platform as a service based on a managed container system, with integrated data service and a powerful ecosystem, for deploying and running modern apps.

Deploy your apps on **Heroku**



Node.js



Ruby



Python



Java



PHP



Go



Scala



Clojure

Heroku Runtime

Heroku runs your apps inside dynos— smart containers on a reliable, fully managed runtime environment. Developers deploy their code written in different frameworks to a build system which produces an app that's ready for execution. The system and language stacks are monitored, patched, and upgraded, so it's always ready and up-to-date. The runtime keeps apps running without any manual intervention.

Heroku Developer Experience (DX)

The HDX is an app-centric approach to software delivery so developers can focus on creating and Continuous Delivery applications, without being distracted by servers or infrastructure. Developers deploy directly from popular tools like Git, GitHub or Continuous Integration (CI) systems. The intuitive web-based Heroku Dashboard makes it easy to manage your app and gain greater visibility into performance.

Data Services and Ecosystem

Heroku Elements let developers extend their apps with Add-ons, customize their application stack with Buildpacks and jumpstart their projects with Buttons. Add-ons are 3rd party cloud services that developers can use to immediately extend their apps with a range of functionality such as data stores, logging, monitoring and more. Heroku provides three fully-managed data service Add-ons: Heroku Postgres, Heroku Redis, and Apache Kafka.

Heroku Operational Experience (OpEx)

The Heroku Operational Experience is a key component of the platform. It helps developers through troubleshooting and remediation of common issues and customizing their ops experience to quickly identify and address negative trends in their application health. Heroku provides a set of tools to alert if something goes wrong, or to automatically scale the web dynos if the response time for web requests exceeds a threshold we specify. Application metrics, Threshold Alerting, and Autoscaling are some of the features you get access to with no extra cost.

Security and Compliance

Heroku regularly performs audits and maintains PCI, HIPAA, ISO, and SOC compliance to further strengthen trust with customers.

Pros/benefits of using Heroku:

- Allows the developer to focus on code instead of infrastructure
- Enhance the productivity of cloud app development team
- Offers single billing for all projects broken down by team
- Support form Modern Open Source Languages
- High-performance Salesforce integration
- Simple Horizontal & Vertical Scalability
- Heroku operation and security team is instantly ready to help you 24/7
- Leading Platform tools and Services Ecosystem
- Helps us to focus on innovation, not operations
- The Heroku Enterprise architecture offers minimal or no downtime during the system updates.
- Fast application lifecycle management and permissions
- A bunch of supportive tools
- Beginner and startup-friend

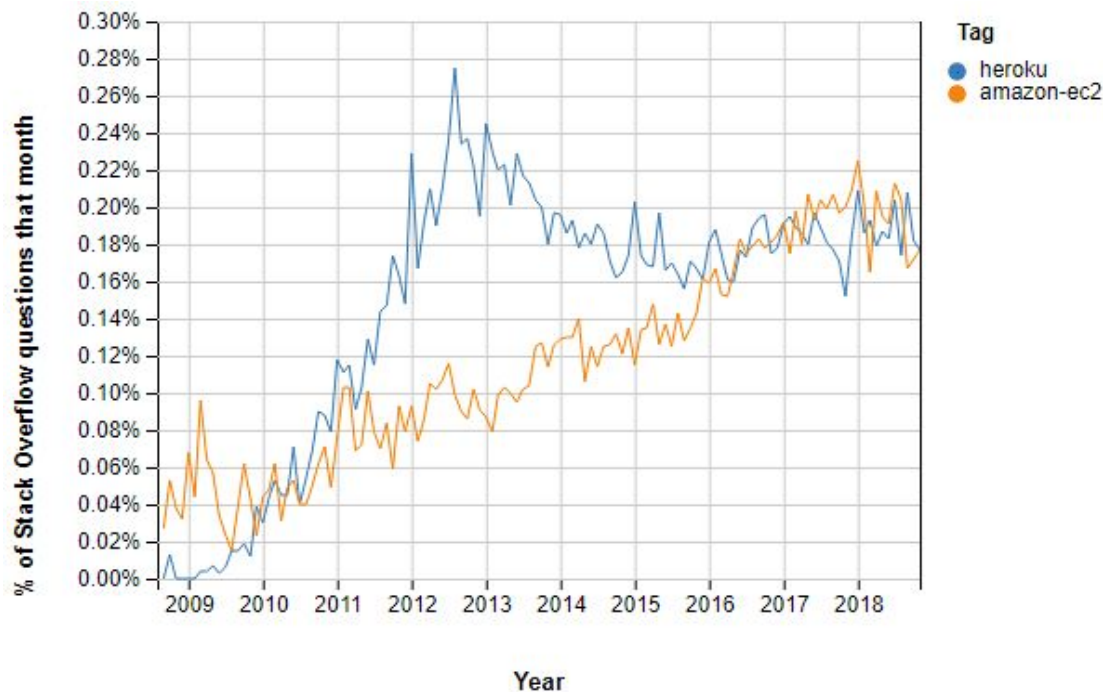
Why enterprise companies trust Heroku

Heroku provides a secure, enterprise-grade platform for organizations of any size. It regularly performs audits and maintains PCI, HIPAA, ISO, and SOC compliance to further strengthen our trust with customers.

Data controls and privacy

Heroku gives us control over our data and which region it's stored, and ensures it remains private.

Heroku Growth over the past few years



AWS Elastic Beanstalk

AWS deploy applications on AWS Elastic Beanstalk by running commands in a Command Line Interface (provided by AWS) or by using the Management Console. After deployment, Elastic Beanstalk manages infrastructure without user's control.

We generally don't need to set up capacity provisioning, load balancing, or scaling, although we're still able to get access to the infrastructure, if necessary, and save multiple configuration options for our application. Elastic Beanstalk uses EC2 instances to host your application, so migrating from AWS Beanstalk to EC2 is easy. And that's great.

Heroku's services are comparable to Amazon's Elastic Beanstalk and they both comes under the category of Platform as a service.

Following will be a brief step wise explanation of what was done to deploy our django server over elastic beanstalk

- To deploy any server using AWS beanstalk service (using django)
We first create a django application on our local machine, after which we create a new folder in the same directory as the django project named ".ebextensions".
- Inside this .ebextensions we create a file named "django.config".
After this make the changes given in the reference link below at the end of this file.
- This will conclude all the modification that we will need to do to start a basic django server on elastic beanstalk
- After this you can simply upload the django project zipped file into a beanstalk environment and access the site after the deployment process finishes

ElasticBeanstalks Environment and other details after deployment has concluded

The screenshot displays the AWS Elastic Beanstalk console interface. The browser address bar shows the URL: [https://console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environment/dashboard?applicationName=CC_project...](https://console.aws.amazon.com/elasticbeanstalk/home?region=us-east-1#/environment/dashboard?applicationName=CC_project_server). The console header includes the AWS logo, a search bar, and the user's profile information (vostartsoft/user1059688=saumya.d18@iits.in @ 9410-8791-8196, N. Virginia, Support).

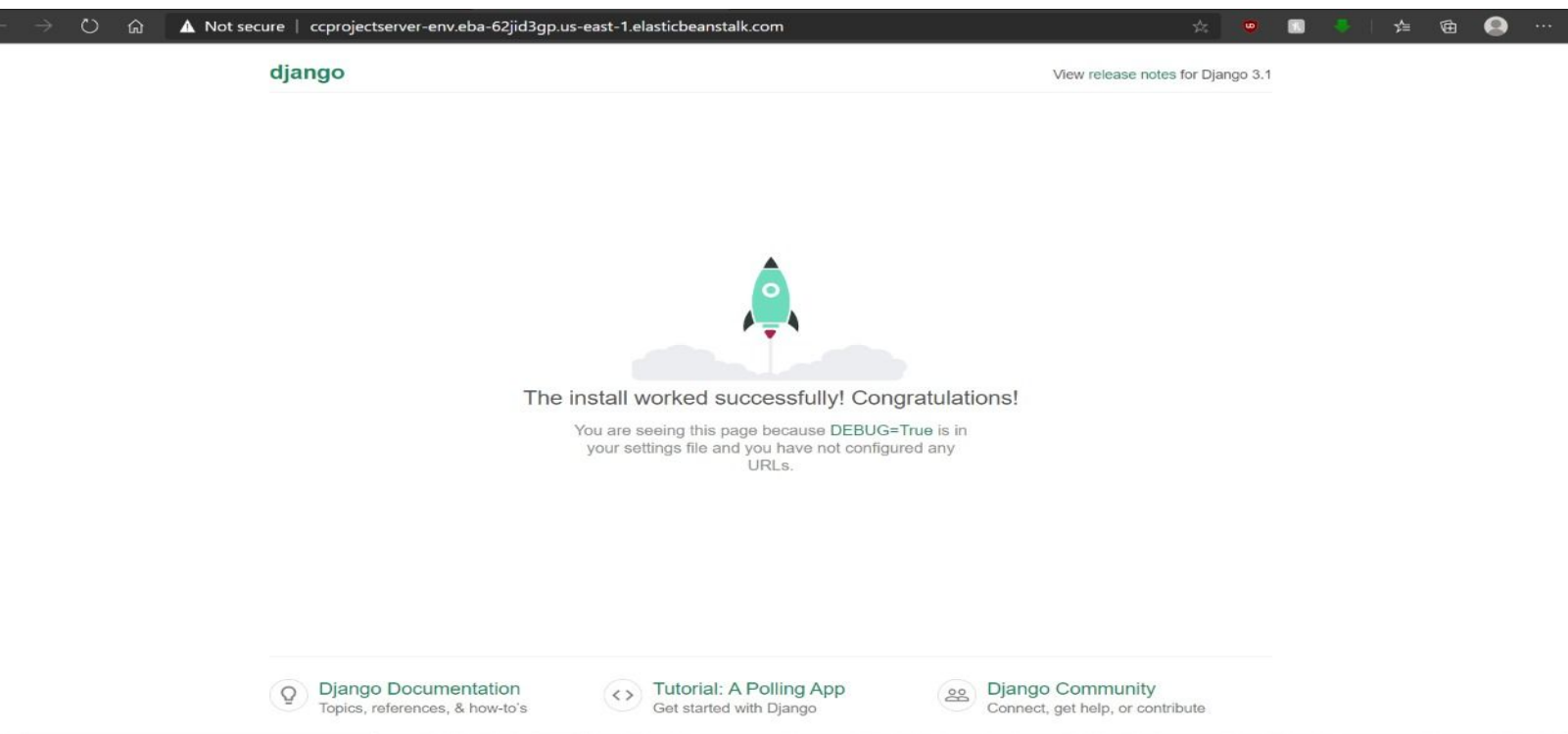
The left sidebar shows the navigation menu with options: Elastic Beanstalk, Environments, Applications, CC_project_server, Application versions, Saved configurations, CcProjectServer-env, Go to environment, Configuration, Logs, Health, Monitoring, Alarms, and Managed updates.

The main content area displays the details for the environment **CcProjectServer-env**. It includes the following information:

- Health:** A green checkmark icon with the text "Ok" and a "Causes" button.
- Running version:** The version is `cc_project_server-source-3`. There is an "Upload and deploy" button.
- Platform:** The platform is `Python 3.7 running on 64bit Amazon Linux 2/3.1.3`. There is a "Change" button.

At the bottom, there is a "Recent events" section with a "Show all" button and a pagination indicator showing 1 event.

Final deployed server



Some out of the box statistics of the deployed server in realtime

Elastic Beanstalk > Environments > CcProjectServer-env > Health

Enhanced health overview

Instances: 1 Total, 1 Ok

[Learn more](#) about enhanced health.

Filter by

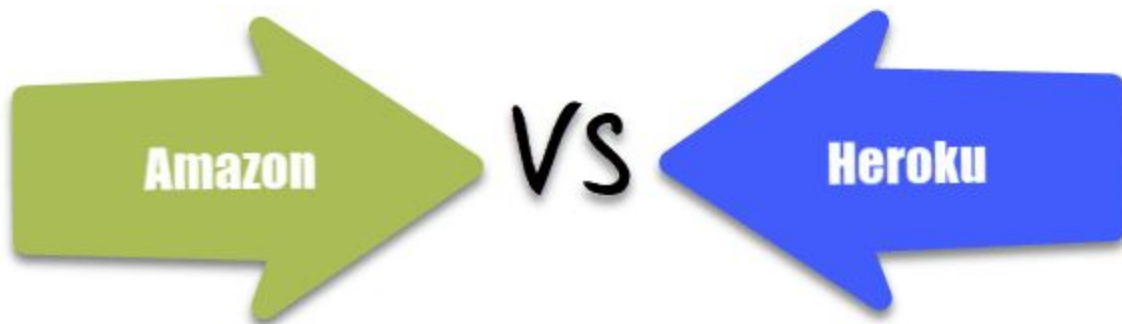
Instance actions

⌂

☒ Auto refresh (11 s)

⊕

Instance ID	Status	Running	Deployment ID	Requests/sec	2xx Responses	3xx Responses	4xx Responses	5xx Responses	P99 Latency	P90 Latency	P75 Latency	
▼ Ok												
Overall	100.0 % of the requests to the ELB are erroring with HTTP 4xx. Insufficient request rate (0.5 requests/min) to determine application health (7 minutes ago).		N/A	N/A	0.2	100%	0.0%	0.0%	0.0%	0.004	0.004	0.004
i-0fab5c5ccdf919856	Ok	8 hours	1	0.2	2	0	0	0	0.004	0.004	0.004	



Parameters	Heroku	AWS
Owner	Salesforce.com	Amazon.com
Hosted on	Amazon's data centers	Proprietary servers IaaS (Amazon EC2 instances)
Type of Service	PaaS	IaaS, PaaS, SaaS
Pricing	Heroku costs \$0.05 per hour.	Starting plan cost you \$0.013 per hour.

Languages	Node.js, Java, Ruby, PHP, Python, Go, Scala, Clojure	.NET, Ruby, NodeJS, Go, Docker, PHP, Python
Features	<ul style="list-style-type: none"> • Fully flexible runtime environment with smart containers (dynos) system. • Allows manual horizontal and vertical scaling • Allows you to roll back your database or code in no time. • App monitoring system to keep track of metrics, like response time, throughput, memory, etc. 	<ul style="list-style-type: none"> • Multiple deployment options and the ability to roll back to the previous version • Quick restart of all app servers by using a single command • Automatic scaling of web apps based on their specific needs and defined conditions.
Best for	Startups, Medium Businesses, Large Enterprises	Medium Businesses, Large Enterprises
Companies using	Toyota, Citrix, Westfield, Yesware, etc.	BMW Group, Airbnb, Coursera, Atlassian, etc.
Complexity	It's software some time too simple, even for professional developers.	Suitable for sophisticated products build by professionals. Not easy for beginners

Built-in Tools for Management and Monitoring	<ul style="list-style-type: none"> • Heroku Command Line • Heroku Application Metrics • Heroku Connect • Heroku Status 	AWS Management Console AWS Command Line Interface (AWS CLI).
Rapid deployment	Heroku offers you a ready-to-use environment which allows you to push your code and make a few configuration changes to get your application running.	The deployment process of AWS service is quite hard.
Need DevOps Engineer	Not at all	Must
Development of server	The creation of a server is a simple process.	The creation of a server is a relatively complicated process.
Computational demands	Low	High

Reference :

[Deploying a Django application to Elastic Beanstalk - AWS Elastic Beanstalk \(amazon.com\)](#)